

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ FİNAL RAPORU**

Afet Durumlarında İHA’ların Etkin Kullanımı İçin Algoritma Araştırması Ve Simulasyon Ortamında Analizi

**PROJE YAZARI**

Ubeydullah Ak ve Umut Engin Berkkan

170421852 Ve 170421044

**DANIŞMAN**

Dr. Eyüp Emre Ülkü

**İstanbul / 2025**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Ubeydullah Ak ve Umut Engin Berkkan’ın “Afet Durumlarında İHA’ların Etkin Kullanımı İçin Algoritma Araştırması Ve Simulasyon Ortamında Analizi” başlıklı bitirme projesi çalışması, …./…./….. tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Dr. Öğr. Üyesi Eyüp Emre Ülkü (Danışman)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

Prof. Dr. Şahin Uyaver (Üye)

Marmara Üniversitesi ......................................................................... (İMZA) ..................

**İÇİNDEKİLER**

Sayfa

KISALTMALAR LİSTESİ………………………………………………………………….i

ŞEKİL LİSTESİ………………………………………………………………….i

TABLO LİSTESİ………………………………………………………………….i

ÖZET……………………………………………………………………………………….iv

BÖLÜM 1. GİRİŞ ............................................................................................................... 1

1.1 Amaç .............................................................................................................................. 2 1.2 Literatür Özeti ................................................................................................................ 3

BÖLÜM 2. MATERYAL VE YÖNTEM ............................................................................ 4

2.1 Materyal ......................................................................................................................... 4 2.1 Donanım ve Çalışma Ortamı …................................................................................. 4 2.2 Robot İşletim Sistemi ................................................................................................... 4 2.3 Programlama Dilleri ve Kütüphaneler ….................................................................. 5 2.4 Coğrafi Konum ve Haritalama .................................................................................. 5 2.5 İHA Modellemesi ...................................................................................................... 5 2.6 Formasyon Tasarımı ve Hedef Belirleme …............................................................. 6 2..7 Otonom Görev Paylaşımı ve Karar Verme ............................................................... 9 2.8 İletişim ve Veri Paylaşımı ........................................................................................ 12 2.9 Optimizasyon Algoritmaları .................................................................................. 13 2.10 Simülasyonların Gerçekleştirilmesi ve Verilerin Analizi ................................... .18 2.11 Senaryo Tanımlamaları ........................................................................................ 19 2.12 Test Prosedürü ve İstatistiksel Yöntemler ........................................................... 25 2.13 Hedefe Ulaşma Süresi ......................................................................................... 25 2.14 Kapsama Yüzdesi ....................................................................................................... 25

BÖLÜM 3. BULGULAR VE TARTIŞMA ........................................................................ 26

3.1 Bulgular ........................................................................................................................ 26 3.2 Tartışma ....................................................................................................................... 27 3.3 Kullanım Senaryoları ................................................................................................ 29 3.4 Sınırlılıklar .................................................................................................................. 29 3.5 Genel Değerlendirme ....................................................................................................30

BÖLÜM 4. SONUÇ ........................................................................................................... 30

4.1 Dinamik Sürü Algoritması ...................................................................................... 31 4.2 Simülasyon Bulguları .............................................................................................. 31 4.3 Kullanılan Teknoloji ................................................................................................ 31 4.4 Gerçek Dünya Testleri ............................................................................................. 31 4.5 Gelişmiş Simülasyon Özellikleri ............................................................................. 32 4.6 Yapay Zeka Entegrasyonu ....................................................................................... 32 4.7 Algoritma İyileştirmeleri ......................................................................................... 32 4.8 Sensör ve Veri Füzyonu ............................................................................................ 32 4.9 Operasyonel Yazılım ve Arayüz ................................................................................... 33

KAYNAKLAR ................................................................................................................... 34

**KISALTMALAR**

**İHA :** İnsansız Hava Aracı

**ROS :** Robot Operating System

**PSO** : Particle Swarm Optimization (Parçacık Sürü Optimizasyonu)

**ACO** : Ant Colony Optimization (Karınca Kolonisi Optimizasyonu)

**GWO** : Grey Wolf Optimizer (Gri Kurt Algoritması)

**WOA** : Whale Optimization Algorithm (Balina Optimizasyonu)

**ŞEKİL LİSTESİ**

**Şekil 1.1** V Formasyon ile yayılmış İHA temsili. ……………………………………..…...6

**Şekil 1.2** Üçgen Formasyon ile yayılmış İHA temsili. …………………………………….7

**Şekil 1.3** Hat Formasyon ile yayılmış İHA temsili. ………………………………………..7

**Şekil 1.4** Grid Formasyon ile yayılmış İHA temsili. ………………………………………8

**Şekil 2.1** İHA’ların PSO algoritması ile yayılmış hali………………………………….....14

**Şekil 2.2** İHA’ların ACO algoritması ile yayılmış hali. ………………………………..…14

**Şekil 2.3** İHA’ların SMA algoritması ile yayılmış hali. ………………………………..…15

**Şekil 2.4** İHA’ların GWO algoritması ile yayılmış hali. ……………………………...…..16

**Şekil 2.5** İHA’ların WOA algoritması ile yayılmış hali. ………………………………….16

**Şekil 2.6** İHA’ların CS algoritması ile yayılmış hali. ……………………………………..17

**TABLO LİSTESİ**

**Tablo 1.1** Temel test senaryosu sonuçları…………………………………………………18

**Tablo 1.2** Çoklu Dağılma………………………………………………………………….19

**Tablo 1.3** Yüksek UAV Sayılı Temel Senaryo…………………………………………….20

**Tablo 1.4** Yüksek UAV Sayılı Çoklu Dağılma…………………………………………... 21

**Tablo 1.5** Tüm Formasyonlarla Tekli Dağılma……………………………………………23

**ÖZET**

Türkiye, coğrafi konumu nedeniyle sıklıkla afetlerle karşılaşan bir ülkedir. Afet durumlarında hızlı ve etkin müdahale, insan hayatını kurtarmak açısından büyük önem taşımaktadır. Bu çalışmada afet bölgelerinde görev yapacak insansız hava araçlarının sürü halinde hareket edebilme yeteneklerinin geliştirilmesi amaçlanmıştır. Çalışmanın temel problemi, mevcut insansız hava araçlarının afet durumlarında sınırlı adaptif yeteneklere sahip olmasıdır. Bu kapsamda, geliştirilen algoritmalar ile İHA’ların sürü zekâsı kullanarak dinamik ve değişken afet senaryolarına adaptif şekilde yanıt vermesi, otonom kararlar alması ve görev paylaşımı yapabilmesi hedeflenmiştir. Algoritmalar, ROS altyapısında simülasyonlarla test edilmiş; veri kaybı, iletişim süresi ve enerji verimliliği gibi kriterler açısından değerlendirilmiştir. Bulgular, önerilen algoritmaların afet yönetiminde kurtarma ve lojistik operasyonları daha hızlı ve verimli hale getirebileceğini göstermiştir. Sonuç olarak, afet durumlarında İHA kullanımına yönelik özgün ve etkin bir algoritma mimarisi geliştirilmiştir.

**ABSTRACT**

Turkey frequently experiences disasters due to its geographical location. Rapid and effective intervention in disaster situations is crucial for saving human lives. This study aims to develop swarm movement capabilities for unmanned aerial vehicles (UAVs) deployed in disaster zones. The primary problem addressed by this research is that existing UAVs possess limited adaptive capabilities in disaster scenarios. Thus, the developed algorithms aim to enhance UAV swarm intelligence, enabling adaptive responses to dynamic and variable disaster conditions, autonomous decision-making, and efficient task allocation. Algorithms were implemented within the ROS framework and tested via simulations, evaluating criteria such as data loss, communication latency, and energy efficiency. Findings demonstrated that the proposed algorithms could significantly enhance rescue and logistical operations during disasters. Consequently, this research offers an original and effective algorithmic architecture for the use of UAV swarms in disaster management.

1. **GİRİŞ**

Türkiye, coğrafi konumu nedeniyle sık sık depremler, sel baskınları ve orman yangınları gibi doğal afetlerle karşı karşıya kalmaktadır. Bu tür olaylarda hızlı ve etkin müdahalede bulunmak, kayıpların en aza indirilmesi açısından kritik önem taşır. Geleneksel arama-kurtarma ve lojistik destek yöntemleri, büyük veya ulaşılması güç alanlarda yetersiz kalabildiği için alternatif teknolojik çözümler üzerine yoğunlaşılmaktadır. İnsansız hava araçları (İHA) özellikle sürü (swarm) halinde görev aldıklarında, geniş ölçekli ve zorlu koşullara sahip afet bölgelerinde hızlı tarama, durum analizi ve operasyon yönetimi yaparak önemli avantajlar sunmaktadır. Türkiye özelinde yapılan bazı çalışmalarda da, İHA’ların afet operasyonlarındaki potansiyeline dikkat çekilmiştir. Örneğin Degirmen ve arkadaşları, gözetleme operasyonlarında İHA’ların rota planlaması için nasıl etkili kullanılabileceğini vurgulamaktadır [3].

* 1. **Amaç**

Bu çalışma, afet durumlarında kullanılmak üzere geliştirilen İHA sürülerinin, otonom görev paylaşımı ve adaptif karar verme yeteneklerini incelemeyi amaçlamaktadır. Projede, Robot Operating System (ROS) altyapısı kullanılarak oluşturulan bir simülasyon ortamında, çoklu İHA’ların koordineli biçimde hareket etmesi, veri paylaşması ve birbirleriyle etkileşim halinde kalması sağlanmıştır. Temel hedef, sürü zekası kavramını İHA’lara entegre ederek dinamik koşullara gerçek zamanlı uyum sağlayabilen ve kurtarma ekiplerinin ihtiyaç duyduğu hayati bilgileri hızlıca aktarabilen bir algoritma mimarisi sunmaktır.

Araştırmanın özgün değeri, mevcut çalışmalara ek olarak İHA’ların sadece veri toplama veya gözlem yapmasının ötesine geçerek kurtarma operasyonlarında koordinasyonu aktif olarak üstlenmesine yönelik yeni bir yaklaşım geliştirmesinden kaynaklanmaktadır. Bu doğrultuda geliştirilen kod ve algoritmalar, farklı afet senaryolarında test edilerek kapsama alanı oranları analiz edilmektedir.

* 1. **Literatür Özeti**

Projemizde önerdiğimiz sürü İHA algoritmaları, mevcut literatürdeki benzer yaklaşımlarla kıyaslandığında birkaç özgün özellik barındırmaktadır. Örneğin Arnold ve arkadaşlarının (2020) çalışmasında, çok rollü İHA sürü davranışları incelenmiş ve geniş alan arama problemlerinde sürü zekasının avantajları gösterilmiştir.[9] Bizim bulgularımız da bu çalışmayı destekler nitelikte olup, özellikle birden fazla iterasyonlu dağılım yaklaşımımız sayesinde alan kapsama oranlarını önemli ölçüde artırdık. Literatürde genellikle tek seferlik arama stratejileri vurgulanırken (örn. spiral ya da rastgele tarama), biz birden fazla aşamalı taramanın daha verimli olabileceğini gösterdik. Bu pratikte şu anlama gelir: Drone’lar bir bölgeyi kabaca taradıktan sonra, elde ettikleri kısmi harita bilgisine göre kendilerini yeniden konumlandırıp kalan boşlukları tarayabilirler. Bu yaklaşımımız, iteratif iyileştirme felsefesiyle literatüre katkı sunmaktadır.

İletişim konusunda, Sun ve arkadaşlarının (2022) çalışmasında belirtilen işbirlikçi hüzmeleme yöntemi, drone sürülerinin iletişimini güçlendirmede kritik bir rol oynuyordu. Biz doğrudan hüzmeleme uygulamamış olsak da, mesafe tabanlı multi-hop iletişim modeli ile benzer bir amacı başardık: Sürü içi ağı koparmadan korumak. Bu, Kumar ve Sonia (2023) tarafından önerilen enerji tasarruf protokolü ile birlikte ele alındığında, sürü haberleşmesinin hem güvenilirlik hem enerji boyutunu optimize etmenin önemli olduğunu ortaya koyuyor.[4][8] Bu çalışmadaki PSO tabanlı çözümde, İHA’lar arasındaki iletişim kopuklukları bir ceza terimi aracılığıyla engellenmiştir. Bu yaklaşım, literatürdeki bazı çalışmalarda olduğu gibi iletişimin sürekliliğini önceliklendirmektedir [5]. Ancak bizim modelimizde enerji etkinliği doğrudan bir iletişim protokolü düzeyinde değil; daha çok konum optimizasyonu yoluyla dolaylı olarak sağlanmıştır. Bu bağlamda, İHA’ların birbirine daha yakın konumlandırılması sayesinde iletişim korunurken, aynı zamanda daha kısa mesafelerde hareket etmeleri nedeniyle enerji tüketimi de azalmıştır [12].

1. **MATERYAL VE YÖNTEM**

Bu bölümde, projenin gerçekleştirilmesinde kullanılan teknik altyapı, geliştirme ortamı, kullanılan yazılım araçları ve İHA modelleme yaklaşımları açıklanmıştır. Simülasyon çalışmaları Ubuntu 22.04 LTS işletim sistemli kişisel bilgisayarlarda gerçekleştirilmiş, ROS 2 Humble sürümü kullanılarak çoklu İHA koordinasyonu sağlanmıştır. Tüm düğümler Python dili ile yazılmış ve veri işleme/görselleştirme için ilgili kütüphaneler entegre edilmiştir. Gerçek dünya koordinatları ile simülasyon ortamı arasında geçiş yapmak için özel bir dönüşüm fonksiyonu (geo\_to\_local) geliştirilmiş, böylece İHA’ların konumları coğrafi anlamda anlamlı hale getirilmiştir. Her bir İHA, bağımsız hareket edebilen sanal bir birim olarak modellenmiş ve farklı formasyonlarda görev yapacak şekilde simüle edilmiştir. Formasyon tipleri (V, delta, düz hat, ızgara) afet bölgesinin yapısına göre dinamik olarak tanımlanabilmekte ve sistem üzerinden seçilebilmektedir.

**2.1. Donanım ve Çalışma Ortamı:**

Proje çalışmaları, araştırmacıların kişisel bilgisayarlarında yürütülmüştür. Geliştirme ve simülasyon çalışmaları için Ubuntu 22.04 LTS işletim sistemi yüklü bilgisayarlar kullanılmıştır. Her iki araştırmacının bilgisayarı da proje boyunca benzer konfigürasyonlara sahip olup, simülasyon işlemlerini gerçek zamanlı olarak çalıştırabilecek işlemci ve bellek kapasitesine sahiptir. Harici özel bir donanım (örneğin gerçek İHA veya ek sensör) kullanılmamış, tüm deneyler yazılım tabanlı simülasyon üzerinden gerçekleştirilmiştir.

**2.2. Robot İşletim Sistemi (ROS 2):**

Simülasyon altyapısını kurmak ve İHA’ların koordinasyonunu sağlamak için ROS 2 (Robot Operating System) framework’ü kullanılmıştır. ROS 2, dağıtık robotik uygulamalar geliştirmek için yaygın olarak kullanılan açık kaynaklı bir middleware’dir. Projede ROS 2 Humble sürümü tercih edilmiştir. ROS 2’nin düğüm-temelli mimarisi sayesinde, her bir İHA sistemde bağımsız birer düğüm (node) olarak modellenebilmiştir. ROS, gerçek zamanlı haberleşme imkânı ve zengin kütüphane desteği ile projemizdeki ihtiyaçlar için ideal bir platform sunmuştur.

**2.3. Programlama Dilleri ve Kütüphaneler:**

Algoritmaların geliştirilmesi ve ROS 2 içerisinde düğüm oluşturmak için Python 3.10 dili kullanılmıştır. Python dilinin ROS 2 ile entegrasyonunu sağlayan rclpy kütüphanesi ile proje kapsamındaki özel ROS düğümü (uav\_node.py) yazılmıştır. Ayrıca matematiksel hesaplamalar ve algoritma implementasyonları için Python’ın standart kütüphaneleri (ör. math, random) kullanıldı. Simülasyon verilerinin kaydedilmesi ve analizinde Python’ın csv modülü ve gerektiğinde numpy, pandas gibi veri işleme kütüphanelerinden yararlanıldı. Görselleştirme amacıyla, ROS’un kendi arayüzü olan Rviz kullanıldı. İHA’ların hareketlerinin 3B ortamda takibi için ROS’un visualization\_msgs/MarkerArray mesaj türü ile anlık gösterimler sağlanmıştır. Rviz sayesinde simülasyon sırasında İHA’ların konumları ve izledikleri yollar görsel olarak doğrulanmıştır.

**2.4. Coğrafi Konum ve Haritalama**

Simülasyon ortamında gerçekçi bir koordinat sistemi kullanmak için coğrafi koordinatlar (enlem-boylam) ile yerel kartezyen koordinatlar arasında dönüşüm yapıldı. Proje kapsamında geo\_to\_local adı verilen yardımcı bir fonksiyon ile belirli bir enlem-boylam-yükseklik değerini metre cinsinden yerel x, y, z koordinatlarına dönüştürmek mümkün kılındı. Bu sayede simülasyondaki İHA konumları, gerçek dünya koordinatlarıyla ilişkilendirilebilmiştir. Örneğin 0.001 derece enlem farkı, yaklaşık 111 metre olarak hesaplanmaktadır; bu tür dönüşümler kod içerisinde otomatik yapılmıştır.

**2.5. İHA Modellemesi**

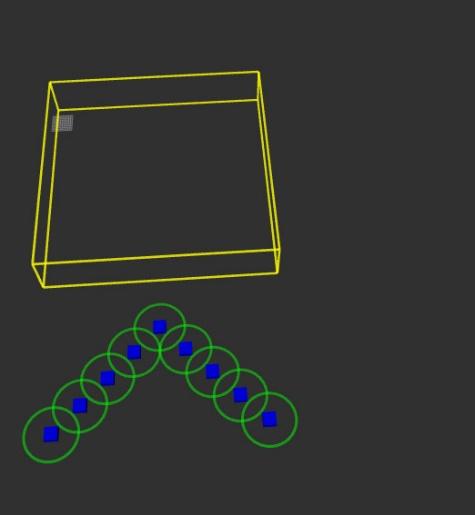
Her bir İHA, belirli bir konuma (x, y, z) sahip, belirli bir iletişim menziline ve kapsama yarıçapına sahip sanal bir araç olarak modellenmiştir. Simülasyon ortamında farklı sayıda İHA tanımlanarak sürü davranışları gözlemlenmiş, formasyon kontrolü ve hedef konumlama gibi görevler üzerinde çeşitli varyasyonlar uygulanmıştır. İHA’ların her biri bağımsız hareket etme ve karar verme kabiliyetine sahiptir; ancak sürü koordinasyonu ve veri paylaşımı ile kolektif hareket stratejileri uygulanmaktadır.

**2.6. Formasyon Tasarımı Ve Hedef Belirleme**

İHA sürüsünün afet bölgesinde gerçekleştireceği tarama görevleri için, İHA’ların alanda nasıl konumlanacağı ve hangi geometrik dizilimde hareket edeceği sistemde tanımlanabilir bir parametre olarak modellenmiştir. Bu kapsamda; V formasyonu, üçgen/Delta dizilimi, düz hat (lineer) formasyonu ve ızgara (grid) dizilimi gibi çeşitli geometrik yapılar kullanıcıya seçenek olarak sunulmuştur. Bu formasyon tipleri, taranacak alanın şekline, genişliğine, engel durumuna ve görev önceliğine göre dinamik olarak belirlenebilir yapıdadır.

**2.6.1. V Formasyonu**

Bir lider İHA’nın önderliğinde, diğer İHA’ların sağ ve sol kanatlarda simetrik bir V harfi şeklinde konumlandığı düzendir. Şekil 1.1 üzerinde de görüldüğü gibi bu formasyon, hareket halindeki sürünün aerodinamik avantaj sağlaması ve belirli bir yöne odaklanarak ilerlemesi için kullanışlıdır. Lider İHA en önde yer alır ve diğer İHA’lar belirli bir açıyla arkasında konumlanır. Bu sayede sürü, geniş bir alan yerine belirli bir hat boyunca ilerlerken birbirine yakın kalır; bu da iletişim kopmalarını azaltabilir.



**Şekil 1.1** V Formasyon ile yayılmış İHA temsili.

**2.6.2. Üçgen/Delta Formasyonu:**

İHA’ların üçgen şeklinde dizildiği bir yapıdır. Şekil 1.2 üzerinde görüldüğü gibi genellikle merkezde bir İHA ve onun etrafında üçgenin köşelerini oluşturan diğer İHA’lar bulunur. Bu formasyon, çevresel kapsama için avantajlıdır; çünkü İHA’lar görüş alanlarını hafifçe farklı yönlere yayarak daha geniş bir açıyı izleyebilirler. Afet bölgesinin orta noktasına odaklanarak dış sınırlarını taramak için delta formasyonu tercih edilebilir.

ekran görüntüsü, daire, çizgi, grafik içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 1.2** Üçgen Formasyon ile yayılmış İHA temsili.

**2.6.3. Düz Hat Formasyonu:**

İHA’ların yan yana düz bir çizgi halinde dizilmesidir. Özellikle geniş alanların paralel şeritler halinde taranması gereken arama senaryolarında etkilidir. Şekil 1.3 üzerinde görüldüğü üzere İHA’lar aralarında sabit mesafelerle bir hat oluşturur ve birlikte ileri doğru hareket ederek altlarındaki alanı tararlar. Bu formasyonun avantajı, bir geçişte geniş bir şeridin taranabilmesidir; ancak hattın iki ucu arasındaki mesafe büyük olursa iletişim ağının kopmaması için dikkatli olunması gerekir.

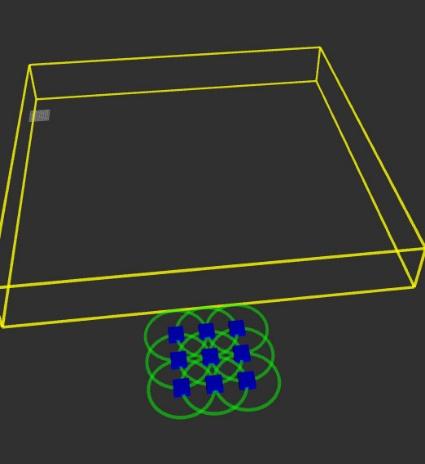
ekran görüntüsü, grafik, daire, çizgi içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 1.3** Hat Formasyon ile yayılmış İHA temsili.

**2.6.4. Izgara Formasyonu:**

İHA’ların bir matriks (ızgara) yapısında çoklu sıra ve kolon şeklinde dizilmesidir. Bu formasyon, şekil 1.4 üzerinde de görüldüğü gibi hem yatay hem dikey eksende çoklu İHA barındırdığı için, belirli bir bölgenin ayrıntılı taranmasında kullanılabilir. Örneğin 4x4’lük bir ızgara formasyonunda 16 İHA, birbirine paralel dört hat üzerinde dörderli gruplar halinde konumlanabilir. Proje kapsamında ızgara formasyonu kavramsal olarak düşünülmüş ancak simülasyonda en fazla 7-8 İHA kullanıldığı için tam anlamıyla büyük bir ızgara dizilimine ihtiyaç duyulmamıştır.



**Şekil 1.4** Grid Formasyon ile yayılmış İHA temsili.

Her bir formasyon tipi, yazılımda parametre olarak tanımlanmış ve merkez noktası etrafında ilgili geometrik kurallara göre İHA konumlarını hesaplayacak fonksiyonlar geliştirilmiştir. Örneğin, V formasyonunda belirlenen bir merkez (lider İHA’nın konumu) etrafında diğer İHA’lar için relatif konum ofsetleri hesaplanır. Üçgen formasyonunda ise bir merkez etrafında katman katman üçgen halkalar oluşturulabilir. Düz hat formasyonunda merkez hattın ortası olacak şekilde İHA’lar yanlara doğru dizilir. Kod içerisinde bu hesaplamalar için compute\_formation\_targets(n, center, ft, spacing) adlı bir fonksiyon tanımlanmıştır. Bu fonksiyon, İHA sayısı n, merkez noktası center, formasyon tipi ft ve İHA’lar arası mesafe spacing parametrelerini alarak her bir İHA için başlangıç pozisyonlarını (Point tipinde) döndürmektedir. Örneğin, ft == 1 (üçgen formasyonu) durumunda fonksiyon bir üçgensel piramit düzeniyle İHA noktalarını oluştururken; ft == 2 (V formasyonu) durumunda bir lider etrafında simetrik kollar oluşturmaktadır. ft == 3 ise düz hat formasyonu olarak kodlanmış, ft parametresi 3 veya daha büyük bir değer aldığında İHA’lar yatay bir çizgi halinde sıraya dizilmektedir. İlerleyen kısımlarda bu fonksiyonun çıktıları simülasyon ortamında görselleştirilmiş ve doğrulanmıştır. Formasyon tasarımı ile ilişkili bir diğer önemli kavram “hold” (bekleme) ve “target” (görev hedefi) noktalarıdır. Sistemde, İHA’ların görev başlamadan önce toplanacağı bekleme noktası (hold center) ve afet bölgesinin merkezi kabul edilen görev hedef noktası (target center) kullanıcı tarafından gerçek coğrafi koordinatlar (enlem, boylam, yükseklik) cinsinden girilebilmektedir. Yazılım, bu gerçek dünya koordinatlarını yerel simülasyon koordinat sistemine dönüştürerek ROS ortamında kullanılabilir hale getirmektedir. Genellikle hold noktası, operasyon güvenliği açısından afet alanının biraz dışında ve yüksekte tanımlanır; İHA’lar simülasyon başında bu noktada seçilen formasyona uygun şekilde dizilir ve kalkış için hazır bekler. Target noktası ise ilgilenilen afet bölgesinin merkezi veya belirli bir görev bölgesidir. Simülasyon senaryosu başlatıldığında, İHA’lar hold pozisyonunda iken sistem “başla” komutu aldığında topluca target bölgesine doğru harekete geçerler. Bu esnada her bir İHA, belirlenen formasyon yapısını koruyarak target noktasına görece kendi hedef alt noktasına ilerler. Örneğin V formasyonunda lider doğrudan merkeze giderken, diğerleri merkezin belirli açılarla sağında/solunda kalacak şekilde yönelir. Görev bölgesine varıldığında, asıl arama-tarama ve kurtarma operasyonu için İHA’ların afet alanına yayılması gerekir. Bu aşamada, kapsama alanını maksimize edecek dağılım algoritmaları devreye girmektedir. Projede, İHA’ların afet bölgesine geldikten sonra etkin bir şekilde alana yayılması ve her birinin uygun bir alt görev bölgesine yönelmesi için çeşitli optimizasyon yöntemleri kullanılmıştır. Özellikle Parçacık Sürü Optimizasyonu (PSO) ve Karınca Kolonisi Optimizasyonu (ACO) gibi meta-sezgisel algoritmalar, İHA’ların afet alanında dağılımını planlamak için adapte edilmiştir. Bu algoritmaların amacı, kapsama boşluğu bırakmadan ve gereksiz kapsama çakışmaları oluşturmadan İHA’ların belirlenen alan üzerinde mümkün olan en geniş alanı taramasını sağlamaktır. PSO yaklaşımıyla, bir parçacık kümesi her bir parçacık tüm İHA’ların konum kombinasyonunu temsil edecek şekilde tasarlanmış; ACO yaklaşımında ise afet alanı hücrelere bölünerek karınca ajanlarının en iyi kapsama kombinasyonunu bulması hedeflenmiştir. Formasyon tasarımı ve hedef atama aşaması, sistemin esnek ve ölçeklenebilir olmasını sağlamıştır. Farklı büyüklükteki afet senaryolarında ve farklı sayılardaki İHA’larla, yukarıda açıklanan formasyon stratejileri başarıyla uygulanarak başlangıç koşulları her senaryoya uygun biçimde yapılandırılabilmiştir.

**2.7. Otonom Görev Paylaşımı Ve Karar Verme**

Simülasyon ortamında her İHA, belirli bir görev döngüsünü takip edecek şekilde tasarlanmış fazlara (durumlara) sahiptir. Bu fazlar, İHA’nın görev süresince hangi aşamada olduğunu belirleyen temel yapıyı oluşturur. Sistem üç ana faza sahiptir: “hold”, “move” ve “final”.

**2.7.1. Hold Fazı**

İHA’ların simülasyon başında bulundukları bekleme durumudur. Bu faz sırasında tüm İHA’lar, bir önceki bölümde belirtilen hold center noktasında, seçilen formasyon geometrisine uygun bir şekilde dizilmiş halde, görevin başlamasını beklerlere. Hold fazında İHA’lar motorlu hareket gerçekleştirmez; yalnızca sabit konumlarını ve formasyon yapılarını korurlar. Bu esnada sistem, tüm İHA’ların kalkış öncesi hazır ve birbirleriyle bağlantıda olup olmadığını kontrol eder. Özellikle bağlantı testleri hold fazında yapılarak İHA’lar arasındaki iletişim kanallarının açık olduğu doğrulanır. Böylece hareket öncesi olası bir iletişim sorunu veya hazırlık eksikliği varsa tespit edilir. Hold fazı, bir bakıma ön yükleme aşamasıdır ve simülasyon kullanıcı komutu ile başlatılana dek devam eder.

**2.7.2. Move Fazı:**

Bu faz, görevin başlatılmasıyla birlikte tüm İHA’ların hedef görev alanına doğru topluca ilerledikleri aktif hareket aşamasıdır. Move fazına geçiş, kullanıcının simülasyonu başlatmasıyla veya önceden belirlenen bir hold süresinin dolmasıyla tetiklenir. Move fazında her bir İHA, global hedef merkezine (yani target center noktasına) doğru, kendine özgü bir hedef nokta belirleyerek ilerler. Bu kendine özgü hedef, formasyondaki pozisyonuna bağlıdır; örneğin V formasyonunda lider direkt merkeze yönelirken diğerleri merkezin belirli offset’leri kadar sağında/solunda olacak noktaları hedefler. İHA’lar hareket ederken formasyon düzenini koruyarak ilerlemeye çalışırlar. Sistem, her bir İHA için kalan mesafeyi dinamik olarak hesaplar ve belirli bir eşik mesafesi (örneğin 1 metre altında) kalana dek move fazını sürdürür. İHA’lardan herhangi biri kendi hedef noktasına çok önce ulaşırsa, diğerlerinin de gelmesini beklemek üzere havada asılı kalır veya yavaşlar, böylece sürü koordineli bir şekilde hareket etmiş olur. Move fazının sonunda, tüm İHA’lar afet bölgesinin tanımlı merkezine yeterince yaklaşmış konuma gelmiş olacaklardır.

**2.7.3. Final Fazı**

Tüm İHA’lar hedeflenen pozisyonlarına başarıyla ulaştığında sistem final fazına geçer. Bu fazda İHA’lar hareketlerini durdurarak uçuşlarını sonlandırır ve planlanan görev icrası tamamlanmış olur. Final fazda sürünün afet alanı üzerindeki son konfigürasyonu “kilitlenir”; bu noktadan itibaren veri analizi ve değerlendirme yapılabilir hale gelir. Özellikle final faz, kapsama kalitesi, iletişim sürekliliği ve görev başarımı gibi metriklerin ölçülmesi için kullanılır. Simülasyon çıktıları bu aşamada kayıt altına alınır. Örneğin kaç İHA’nın hedeflenen alanı kapladığı, herhangi bir kapsanmamış boşluk kalıp kalmadığı veya kaç İHA’nın bağlantısının koptuğu gibi göstergeler final faz çıktılarından hesaplanır.

Belirlenen bu üç temel faz, İHA’ların görev döngüsünü yalın ve anlaşılır biçimde modellemiştir. Her İHA nesnesinin bir phase durumu bulunmakta ve zaman adımları ilerledikçe (simülasyon timer’ı her tetiklendiğinde) İHA’nın bulunduğu faza göre davranışı güncellenmektedir. Kod içerisinde bu mantık timer\_callback fonksiyonu ile düzenli aralıklarla (0.1 saniyede bir) çalıştırılmıştır. Örneğin eğer sistem hold fazında ise ve yeterli bekleme süresi geçtiyse faz durumunu move yapar; move fazında ise her adımda formasyon merkezi global\_form\_center değişkeni hedefe doğru biraz daha yaklaştırılır, İHA’lar da bu merkezi takip ederek ilerlerler. Tüm İHA’lar yeterince hedefe yaklaştığında faz arrived (ulaştı) durumuna alınır ve bir kez konum hesaplaması yapılıp disperse alt fazına geçilir (bkz. 4.4 bölüm). Fazlar arası geçiş koşulları ve hareket güncellemeleri bu şekilde tanımlanarak sürü hareketi düzgün bir şekilde orkestra edilmiştir. Faz mantığı uygulanırken, her bir İHA’nın anlık durumuna göre aldığı aksiyonlar da kodlanmıştır. Örneğin hold fazındaki bir İHA’nın phase değişkeni “hold” değerindeyken konumu sabit tutulur; move fazında phase = "move" olduğunda İHA’nın update() metodu çağrılarak belirli bir hızla hedefe doğru mesafe kat etmesi sağlanır; final fazda ise phase = "final" olup hareket yaptırılmaz. Bu tasarım sayesinde, İHA’ların hem bireysel davranışları hem de toplu hareketleri takip ve kontrol edilebilmiştir. Özetle, otonom görev paylaşımı ve karar verme aşamalarını modellemek için çok durumluluk yaklaşımı benimsenmiştir. Bu yaklaşım, gerçek dünya senaryolarındaki uçuş planlarına da benzer niteliktedir: İHA’lar önce havalanır ve toplanır (hold), ardından göreve uçar (move), görev icra eder ve sonlandırır (final). Uygulanan algoritma ile bu döngü içinde iletişim kontrolü, konum güncelleme ve görev tamamlama süreçleri entegre bir şekilde çalışmaktadır. Sonraki alt bölümde, bu fazlar boyunca İHA’ların kendi aralarında nasıl iletişim kurdukları ve veri paylaştıkları ele alınacaktır.

**2.8. İletişim Ve Veri Paylaşımı**

Bir İHA sürüsünün başarısı, yalnızca bireysel İHA’ların yeteneklerine değil, aynı zamanda ajanlar arası iletişimin güvenilirliğine ve sürekliliğine de bağlıdır. Afet bölgelerinde iletişimin koptuğu durumlarda İHA’ların kendi aralarında ad-hoc ağ kurarak veri iletimi sağlaması kritik önem taşır [1]. Bu nedenle proje kapsamında, İHA’lar arasında sahada sürekli ve sağlıklı bir iletişim kurulmasını sağlayacak bir protokol tanımlanmıştır. Simülasyon ortamında, iletişim modeli için basitleştirilmiş ancak anlamlı bir yaklaşım benimsendi: İki İHA arasındaki mesafenin belirli bir eşiğin altında olması, aralarında doğrudan haberleşme kurulması için yeterli kabul edildi. Bu eşik mesafesi, “iletişim menzili” veya “communication threshold” olarak adlandırılır ve sistem parametresi olarak belirlenir. Öntanımlı olarak 30 metre olarak kullanılan bu eşik, kabaca gerçek dünyadaki RF (radyo frekansı) veya Wi-Fi iletişim menzillerini temsil etmektedir. İletişim protokolü gereği, simülasyon sırasında her zaman her bir İHA çifti arasındaki uzaklık hesaplanır. Eğer herhangi iki İHA’nın arası 30 m veya daha az ise, bu iki İHA’nın birbirine veri gönderebileceği varsayılır ve simülasyonda bu durum için bir “Message Sent” (mesaj gönderildi) durumu kayıt edilir. Aksi halde, yani mesafe 30 m’yi aşıyorsa, “Message NOT sent” çıktısı alınır ve bu iki İHA arasında doğrudan iletişim olmadığı kabul edilir. Bu mekanizma, gerçek dünyadaki sinyal gücü mesafe ilişkisini basitçe modellemektedir; yakın İHA’lar yüksek sinyal alır, uzak İHA’lar bağlantıyı kaybeder. İletişim başarımı bu biçimde ikili (var/yok) olarak ele alınsa da, tüm ağın bir bütün olarak bağlı kalması (sürü içi iletişim sürekliliği) kritik hedeftir. Bunun için simülasyon sırasında arka planda her zaman ağ bağlantı grafı analizi de yapılmıştır. İHA’lar arası mesafe bilgisinden bir komşuluk matrisi oluşturularak, tüm ağın tek bir bileşende (yani herkes en az dolaylı da olsa bağlı) olup olmadığı kontrol edilmiştir. Uygulamada, bir derinlik öncelikli arama (DFS) algoritması ile ağın bağlantılı olup olmadığı hızlıca test edilmiştir; eğer ağda kopuk alt bileşenler varsa, bu durum “iletişim kopukluğu” olarak not edilmiştir. İki İHA doğrudan haberleşemeyecek kadar uzak olsa bile, çoklu sıçrama (multi-hop) iletişim sayesinde ağ bütünlüğü korunmaya çalışılır. Örneğin sürünün en uçtaki iki İHA’sı birbirine 60 m mesafede ve dolayısıyla direkt iletişim kuramıyorsa, aradaki diğer İHA’lar üzerinden iletim gerçekleşebilir. Simülasyon bu durumu, bahsedilen bağlantı grafı üzerinden doğal olarak ele almaktadır; doğrudan mesafe eşiğini aşan İHA’lar, eğer aralarında zincir şeklinde mesafesi uygun İHA’lar varsa, veriyi atlayarak iletebilir. Bu yaklaşım gerçek dünyadaki mesh ağ yapısının basit bir modellemesidir. İletişim protokolünün en kritik yönlerinden biri, iletişim sürekliliğinin optimizasyon süreçlerine dahil edilmesidir. Geliştirilen dağılım algoritmaları (PSO, ACO vb.), konum belirlerken sadece kapsanan alanı değil, aynı zamanda İHA’ların birbirine bağlantılı kalmasını da gözetmektedir. Bunu sağlamak için, optimizasyon algoritmalarında kullanılan uygunluk (fitness) fonksiyonuna bir terim eklenmiştir: Eğer bir çözümdeki İHA konumları ağın kopmasına yol açıyorsa, o çözüme büyük bir ceza puanı verilmektedir. Örneğin PSO algoritmasında, rastgele oluşturulan parçacık pozisyonları arasından sadece tüm grafiği bağlı kılanlar başlangıç için kabul edilmiş; sonraki iterasyonlarda da eğer bir parçacık güncellemesi sonucunda ağ bağlantısı koparsa, o parçacığın uygunluk hesabı atlanmıştır. Benzer şekilde ACO algoritmasında, her yeni İHA pozisyonu seçilirken en azından bir önceki yerleşimdekilerden biriyle bağlantı kuralı aranmış; eğer bu kural sağlanmazsa genele yayılabilecek bir çözüm elde edilemeyeceği düşünülmüştür. Özetle, tanımlanan iletişim protokolü sayesinde sistem, yoğun, düzensiz ve geniş afet bölgelerinde bile İHA’ların bağlantı sürekliliğini sağlama yeteneğini test edebilir hale gelmiştir. Projede, iletişim ile ilgili başarı kriterleri olarak “kaç İHA’nın bağlantısının koptuğu” ve “mesaj iletim oranı” gibi ölçütler kullanılmıştır. Bu ölçütler simülasyon sırasında her zaman sıfır veya %100 olarak kalmamıştır; kimi stres testlerinde, özellikle İHA sayısı azaldığında veya formasyon çok geniş alan kapladığında, iletişim ağı geçici olarak bölünmüştür. Bu gibi durumlar, sonuçlar bölümünde tartışılmıştır. Bununla birlikte genel bulgu, literatürde işbirlikçi hüzmeleme veya benzeri yaklaşımların enerji verimliliğine katkısının yanı sıra iletişim için de kritik olduğudur. Bizim uyguladığımız basit mesafe tabanlı protokol, gerçek sistemlerde daha gelişmiş ağ protokollerine (ör. 5G tabanlı drone ağları) geçiş için temel bir doğrulama sağlamıştır.[8]

**2.9. Optimizasyon Algoritmaları**

İHA sürüsü afet bölgesine ulaştığında, etkili bir arama yapabilmek için kapsam alanını genişletmesi gerekir. Bu, İHA’ların uygun noktalara dağılarak her birinin farklı bir alt bölgeyi taraması anlamına gelir. Rastgele veya sezgisel dağılımlar yerine bu çalışmada optimizasyon algoritmaları kullanılarak mümkün olan en iyi dağılım hedeflenmiştir. Projede toplam altı farklı optimizasyon yaklaşımı ele alınmıştır: Parçacık Sürü Optimizasyonu (PSO), Karınca Kolonisi Optimizasyonu (ACO), Sürü Hareket Algoritması (SMA - boids temelli), Gri Kurt Optimizasyonu (GWO), Balina Optimizasyonu Algoritması (WOA) ve Guguk Kuşu Arama (Cuckoo Search - CS). Bu algoritmalar sırasıyla kod içerisinde pso\_distribute(), aco\_distribute(), sma\_distribute(), gwo\_distribute(), woa\_distribute() ve cuckoo\_distribute() fonksiyonlarıyla uygulanmıştır.

**2.9.1. Parçacık Sürü Optimizasyonu (PSO):**

PSO algoritması, sürekli optimizasyon problemlerine etkili bir çözüm sağlar. Projede, İHA’ların konumlandırma problemi parçacık sürüsü mantığıyla çözülmüştür şekil 2.1 üzerinde dağılım gözlemlenebilmektedir. Her parçacık, tüm sürünün potansiyel konumlandırmasını temsil eder ve fitness fonksiyonu ile uygunluğu değerlendirilir. Fitness fonksiyonu; kapsama alanlarının çakışması, görev alanı dışına çıkma ve iletişim ağının sürekliliği kriterlerini içerir. Her iterasyon sırasında parçacıklar, kendi buldukları en iyi çözümü (pbest) ve sürünün bulduğu en iyi genel çözümü (gbest) temel alarak konumlarını günceller.

renklilik, grafik, sanat, kalıp, desen, düzen içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.1** İHA’ların PSO algoritması ile yayılmış hali.

**2.9.2 Karınca Kolonisi Optimizasyonu (ACO):**

ACO algoritması genellikle ayrık optimizasyon problemlerinde kullanılır. Bu çalışmada görev alanı belirli bir grid yapısına bölünmüş ve her karınca, bu grid üzerindeki hücrelerden uygun konumları seçmiştir. Seçim işlemi, İHA'ların iletişim ağını koruyacak şekilde yapılmıştır. Her iterasyon sonunda feromon güncellemesi yapılarak çözüm kalitesi iyileştirilir. İyi çözümler feromon değerlerini artırır, böylece daha sonraki iterasyonlarda bu hücreler daha fazla tercih edilir. Şekil 2.2 üzerinde de görüldüğü gibi İHA’lar alana etkin bir biçimde dağılabilmektedir.

renklilik, daire, sanat, grafik içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.2** İHA’ların ACO algoritması ile yayılmış hali.

**2.9.3. Sürü Hareket Algoritması (SMA - Boids Temelli):**

Boids algoritması, global optimizasyondan ziyade yerel kurallarla sürü davranışlarını simüle eder. Her İHA, komşularının konum ve hareket bilgisine göre separation (ayrışma), alignment (hizalanma) ve cohesion (uyum) vektörlerini hesaplayarak konumlarını sürekli günceller. Şekil 2.3 bize dağılımın son halini göstermektedir. Bu algoritmada, her İHA kendi lokal çevresindeki durumu değerlendirerek konumunu belirler ve sürü içinde güvenli mesafeleri koruyarak çarpışmaları önler. Bu yöntem hızlı ve güvenli dağılım sağlamakta etkili olmakla birlikte, global optimuma ulaşma garantisi sunmaz.

daire, renklilik, grafik, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.3** İHA’ların SMA algoritması ile yayılmış hali.

**2.9.4 Gri Kurt Optimizasyonu (GWO):**

GWO algoritması, gri kurtların sürü halinde avlanma davranışlarını temel alarak optimum çözümler arar. Bu algoritmada kurtlar sürüsü, avlanma sırasında alfa, beta ve delta olarak belirlenen lider kurtları takip eder. Her İHA, liderlerin konumlarını referans alarak hareket eder ve iterasyonlar boyunca bu liderlere yakınlaşarak çözüm uzayında hızlıca optimal çözüme yakınsar. GWO, şekil 2.4 üzerinde de görüldüğü üzere hızlı ve etkin çözümler üretir.

grafik, sanat, renklilik, ekran görüntüsü içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.4** İHA’ların GWO algoritması ile yayılmış hali.

**2.9.5. Balina Optimizasyonu Algoritması (WOA):**

WOA algoritması, balinaların avlanma davranışları ve spiral hareketlerini modelleyerek çözüm bulur. İHA’lar, bu algoritma kapsamında sürekli olarak hedef konumlarını güncelleyerek avlanma benzeri spiral hareketler gerçekleştirir. Şekil 2.5 üzerinde görüldüğü üzere iletişim menzilinden çıkmadan alan üzere yayılmışlardır. Her iterasyonda balinaların avlanma hareketleri simüle edilerek çözümler iyileştirilir. Bu algoritma, karmaşık optimizasyon problemlerinde hızlı yakınsama yeteneği ile öne çıkar.

daire, renklilik, sanat, grafik içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.5** İHA’ların WOA algoritması ile yayılmış hali.

**2.9.6. Guguk Kuşu Arama (Cuckoo Search - CS):**

CS algoritması, guguk kuşlarının yuva seçme ve yumurta bırakma davranışlarından esinlenir. Algoritma, mevcut çözüm kümelerini iteratif olarak rastgele değiştirir ve uygun olmayan çözümleri hızla eleyerek en iyi çözümleri muhafaza eder. Bu durum şekil 2.6 üzerinde de gözlemlenebilmektedir.Bu yaklaşım geniş arama alanlarını etkin şekilde tarama özelliğine sahiptir ve yerel optimumlarda takılma ihtimalini azaltır. İHA’ların görev alanına dağılımında hızlı ve etkin sonuçlar sağlayabilir.

sanat, kalıp, desen, düzen, renklilik, daire içeren bir resim

Yapay zeka tarafından oluşturulmuş içerik yanlış olabilir.

**Şekil 2.6** İHA’ların CS algoritması ile yayılmış hali.

Bu altı farklı algoritma, İHA’ların afet bölgesindeki optimal dağılımını farklı perspektiflerden ele alarak çözüm üretmiştir. Her algoritmanın sonuçları detaylı simülasyon analizleri ile incelenmiş ve algoritmaların avantajları, sınırlamaları, performans kriterleri ve farklı senaryolarda uygulanabilirlikleri kapsamlı olarak tartışılmıştır.

**2.10 Simulasyonların Gerçekleştirilmesi ve Verilerin Analizi**

Simülasyon ortamı, 150 m x 150 m boyutlarında sanal bir afet bölgesi olarak tanımlanmıştır. Bu bölge, bir deprem sonrası arama-kurtarma operasyonunun yapılacağı kentsel bir alanı sembolize etmektedir. Her bir senaryoda, belirli sayıda İHA (genellikle 5, 6 veya 12 olarak değiştirildi) başlangıçta afet alanının dışında bir bekleme noktası üzerinde belirli bir formasyonla konumlandırılmıştır. Senaryo parametreleri arasında şu değişkenler yer almıştır:

**2.10.1 İHA Sayısı**

5,9 ve 12 kadar olmak üzere farklı İHA sayıları ile testler gerçekleştirilmiştir. Az sayıda İHA ile sınırlı kaynak senaryoları, çok sayıda İHA ile de yoğun sürü senaryoları incelenmiştir.

**2.10.2 Formasyon Tipi**

V, üçgen (delta) ve düz hat formasyonları, farklı başlangıç dizilimlerini temsil etmek üzere senaryolara dahil edilmiştir. Örneğin bir senaryoda 5 İHA V formasyonunda başlatılırken başka bir senaryoda 5 İHA düz hat halinde başlamıştır. Bu sayede başlangıç diziliminin sonuçlar üzerindeki etkisi gözlemlenmiştir.

**2.10.3. Algoritma Seçimi**

Dağılım için kullanılan algoritma (PSO, ACO veya boids) her senaryoda ayrı ayrı seçilmiştir. Böylece aynı koşullar altında farklı algoritmaların performansı karşılaştırılabilmiştir. Örneğin 5 İHA’lı bir senaryo hem PSO ile hem ACO ile yürütülüp sonuçlar kaydedilmiştir.

**2.10.4 İletişim Menzili ve Kapsama Yarıçapı**

Baz senaryolarda iletişim menzili 50 m olarak kullanıldı (bazılarında 30 m de denendi), her bir İHA’nın sensör kapsama yarıçapı ise 20 m olarak alındı. Bu değerler literatüre uygun olarak seçilmiştir [6]. Ancak bazı stres testlerinde iletişim eşiği yapay olarak düşürülerek kopukluk senaryoları incelenmiştir (örneğin 20 m menzil ile de denemeler yapılmıştır) [7]

**2.11 Senaryo**

Bu bölümde, belirli parametrelerle oluşturulmuş beş senaryo tanımlanmış ve her biri için tüm algoritmalar test edilmiştir. Senaryoların amacı, farklı İHA sayıları, formasyon tipleri, alan genişlikleri ve dağılım iterasyonları gibi değişkenlerin sistem performansına etkisini analiz etmektir.

**2.11.1 Senaryo 1 – Temel Test Senaryosu**

Bu senaryoda 5 İHA, Delta formasyonunda tek dağılma şeklinde tüm algoritmalarla test edilmiştir. Alan genişliği 50 metre, kapsama yarıçapı 10 metre ve iletişim menzili 20 metre olarak belirlenmiştir.

| **Zaman (saniye)** | **UAV Sayısı** | **Algoritma** | **Formasyon** | **Kapsama (%)** |
| --- | --- | --- | --- | --- |
| 4.56 | 5 | PSO | Delta | 48.96 |
| 8.42 | 5 | ACO | Delta | 48.80 |
| 5.59 | 5 | SMA | Delta | 40.16 |
| 9.25 | 5 | GWO | Delta | 47.68 |
| 8.09 | 5 | WOA | Delta | 33.76 |
| 8.09 | 5 | CO | Delta | 52.64 |

**Tablo 1.1** Temel test senaryosu sonuçları.

Tüm algoritmalar test edildiğinde, CO algoritması %52.64 ile en yüksek kapsama oranını sağlamıştır. PSO ve ACO benzer sonuçlar verirken, WOA algoritmasının kapsama başarısı düşüktür. Tüm testlerde formasyon sabit tutulmuş, sadece algoritmalar karşılaştırılmıştır.

**2.11.2- Senaryo 2 – Çoklu Dağılma**

Bu senaryoda 5 İHA, Delta formasyonunda 3 iterasyonlu çoklu dağılma şeklinde tüm algoritmalarla test edilmiştir. Alan genişliği 100 metre, kapsama yarıçapı 10 metre ve iletişim menzili 20 metre olarak belirlenmiştir.

| **Zaman (saniye)** | **UAV Sayısı** | **Algoritma** | **Formasyon** | **Kapsama (%)** |
| --- | --- | --- | --- | --- |
| 5.29 | 5 | PSO | Delta | 15.76 |
| 16.28 | 5 | PSO | Delta | 29.32 |
| 13.25 | 5 | PSO | Delta | 41.08 |
| 11.32 | 5 | ACO | Delta | 11.84 |
| 11.91 | 5 | ACO | Delta | 19.00 |
| 4.22 | 5 | ACO | Delta | 21.52 |
| 6.80 | 5 | SMA | Delta | 10.36 |
| 7.90 | 5 | SMA | Delta | 17.04 |
| 6.89 | 5 | SMA | Delta | 25.88 |
| 8.10 | 5 | GWO | Delta | 15.36 |
| 11.60 | 5 | GWO | Delta | 24.72 |
| 12.70 | 5 | GWO | Delta | 37.84 |
| 6.10 | 5 | WOA | Delta | 12.96 |
| 7.80 | 5 | WOA | Delta | 20.96 |
| 6.40 | 5 | WOA | Delta | 24.68 |
| 7.78 | 5 | CO | Delta | 15.08 |
| 6.18 | 5 | CO | Delta | 18.80 |
| 5.28 | 5 | CO | Delta | 22.52 |

**Tablo 1.2** Çoklu Dağılma

Bu senaryoda 3 iterasyon boyunca test edilen algoritmalar arasında en yüksek kapsama oranı %41.08 ile PSO algoritmasının 3. iterasyonunda elde edilmiştir. GWO algoritması da %37.84'e ulaşarak dikkat çekici bir performans göstermiştir. SMA, WOA ve CO algoritmaları %20–26 aralığında kapsama başarısı sergilemiş, ACO ise genellikle %12–22 aralığında kalarak en düşük performansı göstermiştir. Çoklu dağılma, özellikle PSO ve GWO algoritmaları için kapsama yüzdesini anlamlı biçimde artırmıştır.

**2.11.3- Senaryo 3 – Yüksek UAV Sayılı Temel Senaryo**

Bu senaryoda 12 İHA, Delta formasyonunda tek seferlik dağılma ile test edilmiştir. Alan genişliği 100 metre, kapsama yarıçapı 10 metre ve iletişim menzili 20 metre olarak belirlenmiştir.

| **Zaman (saniye)** | **UAV Sayısı** | **Algoritma** | **Formasyon** | **Kapsama (%)** |
| --- | --- | --- | --- | --- |
| 11.11 | 12 | PSO | Delta | 32.80 |
| 18.49 | 12 | ACO | Delta | 28.24 |
| 7.10 | 12 | SMA | Delta | 16.64 |
| 17.33 | 12 | GWO | Delta | 33.68 |
| 7.20 | 12 | WOA | Delta | 23.92 |
| 10.92 | 12 | CO | Delta | 24.80 |

**Tablo 1.3** Yüksek UAV Sayılı Temel Senaryo

Bu senaryoda GWO algoritması %33.68 ile en yüksek kapsama oranını elde etmiştir. PSO algoritması da %32.80 ile benzer bir başarı göstermiştir. ACO, CO ve WOA algoritmaları orta düzeyde performans sergilerken, SMA algoritması %16.64 ile en düşük kapsama oranında kalmıştır. 12 İHA ile gerçekleştirilen bu tek seferlik dağılım senaryosu, algoritmaların yüksek sayıda araçla nasıl performans gösterdiğini ortaya koymuştur.

**2.11.4- Senaryo 4 – Yüksek UAV Sayılı Çoklu Dağılma**

Bu senaryoda 12 İHA, Delta formasyonunda 3 iterasyonlu çoklu dağılma yöntemiyle tüm algoritmalarla test edilmiştir. Alan genişliği 100 metre, kapsama yarıçapı 10 metre ve iletişim menzili 20 metre olarak belirlenmiştir.

| **Zaman (saniye)** | **UAV Sayısı** | **Algoritma** | **Formasyon** | **Kapsama (%)** |
| --- | --- | --- | --- | --- |
| 21.35 | 12 | PSO | Delta | 33.04 |
| 14.57 | 12 | PSO | Delta | 43.20 |
| 14.13 | 12 | PSO | Delta | 62.40 |
| 14.20 | 12 | ACO | Delta | 28.92 |
| 13.28 | 12 | ACO | Delta | 45.72 |
| 11.90 | 12 | ACO | Delta | 55.40 |
| 6.10 | 12 | SMA | Delta | 18.92 |
| 8.20 | 12 | SMA | Delta | 29.48 |
| 5.80 | 12 | SMA | Delta | 32.88 |
| 16.39 | 12 | GWO | Delta | 32.80 |
| 14.97 | 12 | GWO | Delta | 47.32 |
| 15.16 | 12 | GWO | Delta | 62.68 |
| 7.50 | 12 | WOA | Delta | 19.28 |
| 8.80 | 12 | WOA | Delta | 30.36 |
| 7.50 | 12 | WOA | Delta | 32.60 |
| 13.03 | 12 | CO | Delta | 24.56 |
| 7.63 | 12 | CO | Delta | 26.20 |
| 7.42 | 12 | CO | Delta | 27.36 |

**Tablo 1.4** Yüksek UAV Sayılı Çoklu Dağılma

Bu senaryoda 3 iterasyonlu dağılımla GWO algoritması %62.68 kapsama oranıyla en başarılı sonucu vermiştir. PSO ve ACO da sırasıyla %62.40 ve %55.40 ile yüksek başarı sağlamıştır. SMA, WOA ve CO algoritmaları önceki senaryolara kıyasla daha düşük ama artan bir performans göstermiştir. Çoklu dağılma, genel olarak tüm algoritmalar için kapsama başarısını artırmıştır.

**2.11.5 Senaryo 5 Tüm Formasyonlarla Tekli Dağılma**

Bu senaryoda 6 İHA, dört farklı formasyon tipi (Delta, V, Grid, Line) ile tek dağılma şeklinde test edilmiştir. Alan genişliği 50 metre, kapsama yarıçapı 10 metre ve iletişim menzili 20 metre olarak belirlenmiştir.

| **Zaman (saniye)** | **UAV Sayısı** | **Algoritma** | **Formasyon** | **Kapsama (%)** |
| --- | --- | --- | --- | --- |
| 7.15 | 6 | PSO | Delta | 60.80 |
| 13.35 | 6 | PSO | V | 60.64 |
| 6.55 | 6 | PSO | Grid | 57.60 |
| 8.85 | 6 | PSO | Line | 51.84 |
| 7.89 | 6 | ACO | Delta | 68.64 |
| 6.87 | 6 | ACO | V | 57.12 |
| 5.76 | 6 | ACO | Grid | 59.68 |
| 8.76 | 6 | ACO | Line | 58.72 |
| 7.17 | 6 | SMA | Delta | 55.36 |
| 11.68 | 6 | SMA | V | 51.52 |
| 6.58 | 6 | SMA | Grid | 57.28 |
| 7.76 | 6 | SMA | Line | 54.40 |
| 4.83 | 6 | GWO | Delta | 56.80 |
| 8.63 | 6 | GWO | V | 51.52 |
| 7.83 | 6 | GWO | Grid | 56.64 |
| 9.63 | 6 | GWO | Line | 48.00 |
| 6.90 | 6 | WOA | Delta | 50.88 |
| 8.89 | 6 | WOA | V | 50.72 |
| 5.99 | 6 | WOA | Grid | 50.40 |
| 6.39 | 6 | WOA | Line | 47.52 |
| 6.35 | 6 | CO | Delta | 62.40 |
| 10.55 | 6 | CO | V | 63.68 |
| 5.35 | 6 | CO | Grid | 67.68 |
| 6.15 | 6 | CO | Line | 65.12 |

**Tablo 1.5 Tüm Formasyonlarla Tekli Dağılma**

Bu senaryoda tüm algoritmalar ve formasyonlar test edilmiştir. ACO algoritması Delta formasyonunda %68.64 ile en yüksek kapsama başarısını elde etmiştir. CO algoritması da Grid formasyonunda %67.68 ve Line formasyonunda %65.12 ile yüksek performans göstermiştir. Genel olarak, ACO ve CO algoritmaları çeşitli formasyonlarda yüksek kapsama sağlamıştır. WOA ise tüm formasyonlarda %50'nin biraz üzerinde kalarak en düşük performans gösteren algoritmalardan biri olmuştur. Bu senaryo, formasyonun ve algoritma seçiminin kapsama oranı üzerindeki etkisini net biçimde ortaya koymaktadır.

**2.12 Test Prosedürü ve İstatistiksel Yöntemler**

Proje kapsamında gerçekleştirilen simülasyonların bilimsel açıdan geçerli ve güvenilir sonuçlar üretmesi için titiz bir test prosedürü uygulanmıştır. Öncelikle, her bir senaryo birden fazla tekrarla çalıştırılmıştır. Her bir konfigürasyon (örneğin Senaryo A gibi) en az 5, en çok 10 tekrar koşturulmuştur.

Her bir tekrar, simülasyonun başlangıcında İHA’ların rastgele ufak konum sapmaları ve/veya rastgele dahili algoritma inisiyatifleri (PSO ve ACO için rastgelelik) nedeniyle farklılık yaratmıştır. Bu sayede, tek bir koşul için birden çok örneklem toplanmış ve istatistiksel anlamlılık sağlanmıştır. Örneğin Senaryo A, 5 kez tekrarlanmış ve her birinde PSO algoritmasının başlangıç rastgeleliği farklı bir dağılım vermiştir; sonuç olarak kapsama oranı bu simülasyonların ortalaması olarak alınmıştır. Tekrarlı çalıştırmalar, rastlantısal varyasyonların etkisini azaltmış ve sistem davranışının ortalama eğilimi daha doğru biçimde gözlemlenmiştir.

Simülasyon sonuçlarının analizi için çeşitli performans metrikleri tanımlanmıştır. Bunlar daha önceki bölümlerde de bahsi geçen:

**2.13 Hedefe Ulaşma Süresi**

İHA’ların hold noktasından afet bölgesine varması için geçen süre (saniye cinsinden). Sürü halindeki tüm İHA’ların hedef konumlarına varmış olması ile ölçülür (genelde son İHA baz alınır). Bu süre, müdahale hızının bir göstergesidir.

**2.14 Kapsama Yüzdesi**

İHA’ların sensör/kamera kapsama alanlarının birleştirilmesiyle elde edilen toplam alanın, görev bölgesi toplam alanına oranı (%). Afet alanının ne kadarının tarandığını gösterir. Bu hesap, İHA’ların kapsama disklerinin birleşiminin alanının, kare şeklindeki toplam alanına bölünmesiyle elde edilmiştir. Kapsama hesaplanırken üst üste binen alanlar tek sayılmıştır (bir alan 2 İHA tarafından görülse de bir kere sayıldı) ve kapsama dışı kalan boşluklar yüzdelik olarak ifade edilmiştir.

Bu metriklerin her biri için temel istatistiksel analizler yapılmıştır: Ortalama değerler esas alınarak tablolar oluşturulmuştur. Tablolarda verilen sonuçlar, tekrarlı çalıştırmaların ortalamasını temsil etmektedir. Dağılım algoritmalarının performansları da bu ortalamalar üzerinden kıyaslanmıştır. Simülasyonlarda bazı algoritmaların başlangıç duyarlılığı nedeniyle varyans gösterebildiği gözlemlenmiştir. Bu gibi durumlarda, örneklem sayısı artırılarak daha güvenilir bir ortalama elde edilmeye çalışılmıştır.

Genel olarak, PSO ve ACO gibi algoritmaların benzer koşullarda küçük farklarla birbirinden ayrıldığı, Boids temelli SMA algoritmasının ise başlangıç koşullarına daha hassas tepkiler verdiği görülmüştür. Bu bulgular tartışma bölümünde yorumlanmıştır.

Tüm bu yöntemsel çerçeve sayesinde, sürü halindeki İHA’ların dinamik afet ortamlarında nasıl hareket edeceği, hangi iletişim stratejilerini kullanacağı ve hangi parametrelerin sonuçlar üzerinde ne derece etkili olduğu nicel verilerle ortaya konulmuştur. Böylece projenin başındaki hipotez ve hedefler, kapsamlı bir deneysel analizle doğrulanmaya çalışılmıştır.

1. **BULGULAR VE TARTIŞMA**

Bu bölümde, geliştirilen simülasyon altyapısının performansı ve literatürdeki benzer çalışmalarla karşılaştırması yapılmıştır. Optimizasyon algoritmalarının (PSO, ACO vb.) karşılaştırmalı başarıları değerlendirilmiş, hibrit yöntemlerin potansiyeli vurgulanmıştır. Gerçek dünya uygulamalarındaki zorluklar ve sınırlılıklar, simülasyonun basitleştirilmiş koşullarıyla birlikte ele alınmıştır. Ayrıca, proje altyapısının genişletilebilirliği ve yeni algoritmalara (GWO, WOA, CS) açık yapısı ifade edilmiştir. Elde edilen sonuçlar, otonom görev dağılımı ve dağıtık kontrol konularında özgün katkılar sunmuş; afet sonrası arama-kurtarma ve acil iletişim gibi senaryolarda pratik fayda sağlayabileceği gösterilmiştir. Bununla birlikte, İHA sayısının azlığı, parametre hassasiyetleri ve simülasyonun sınırlı gerçekçilikte olması gibi bazı kısıtlar da tespit edilmiştir. Tüm bulgular değerlendirildiğinde, çalışma hem akademik hem de uygulamalı bağlamda önemli bir başlangıç noktası sunmaktadır.

* 1. **Bulgular**

Bu proje kapsamında, afet bölgelerinde çoklu İHA (İnsansız Hava Aracı) sürülerinin etkili biçimde koordine olmasını sağlayacak bir simülasyon altyapısı ROS aracılığı ile geliştirilmiştir. İHA sürülerinin afet yönetiminde kullanımıyla ilgili mevcut literatür incelenmiş ve farklı formasyon (V, üçgen/Delta, düz hat vb.) ile iletişim protokollerinin avantajları ve dezavantajları belirlenmiştir.[6] Bu sayede, projede uygulanacak yaklaşımlar için kapsamlı bir temel oluşturulmuştur. Robot Operating System (ROS) ve Python dili kullanılarak bir simülasyon ortamı yapılandırılmıştır. Bu ortamda, her bir İHA’nın konumu, iletişim menzili, hareket hızı ve otonom davranışları (örneğin “hold”, “move”, “final” fazları) kontrol edilip gözlemlenebilmektedir. Projenin çekirdeğini oluşturan ilgili Python dosyasında, İHA’ların sürü halinde hareket ederken birbirleriyle mesajlaşma ve belirli bir hedefe doğru toplu biçimde ilerleme yetenekleri kodlanmıştır. İHA’lar, iletişim eşiği (örneğin 30 metre) dahilinde kaldığında veri paylaşımı yapabilmekte, aksi halde bağlantı kopukluğu yaşamaktadır.

**3.2** **Tartışma**

Bu bölümde elde edilen sonuçlar literatürdeki çalışmalar ile karşılaştırılmakta, projenin sınırlılıkları ve güçlü yönleri tartışılmaktadır. Ayrıca geliştirilen yöntemlerin pratik uygulamaları ve iyileştirme alanları üzerinde durulmaktadır.

**3.2.1 Algoritmaların Birlikte Çalışması ve Entegrasyon**

Projemizde her bir optimizasyon algoritması bağımsız şekilde çalıştırılmıştır. Oysa gelecekte, farklı algoritmaların hibrit kullanımı da mümkün olabilir. Örneğin Zhou ve arkadaşlarının önerdiği hibrit ACO-PSO yöntemi [6], PSO’nun hızlı küresel arama yeteneği ile ACO’nun yerel iyileştirme ve bağlantı koruma başarımını birleştirmeyi amaçlamaktadır. Bunlara ek olarak, literatürde Penguin Optimization Algorithm gibi doğadan ilham alan alternatif yaklaşımlar da arama-kurtarma görevlerinde test edilmiştir [10]. Bu algoritmalar, çoklu robot sistemlerinde etkili alan taraması sağlama konusunda gelecek vaat eden stratejiler sunmaktadır. Bizim simülasyon sonuçlarımızda PSO algoritması daha yüksek kapsama oranı sağlarken, ACO algoritması İHA’lar arası bağlantının korunmasında daha başarılı olmuştur [4], [12]. Bu bağlamda, bir hibrit yaklaşım bu iki yöntemin güçlü yönlerini bir araya getirerek daha dengeli bir çözüm sunabilir. Tartışmanın bu noktasında, önerdiğimiz mimarinin bu tür geliştirmelere açık olduğunu belirtmek gerekir. Kod altyapısında algorithm parametresi için 4, 5 ve 6 gibi boş yerler bırakmamızın amacı da budur: İleride Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA) veya Cuckoo Search (CS) gibi algoritmaların da test edilmesi hedeflenmektedir. GWO’nun sürü içi lider takibi konsepti formasyon koruma problemi ile uyumlu olabilir [4]; WOA’nın avcı–balina yaklaşımı, kapsanmayan bölgelerin “avlanması” için etkili olabilir [11]; CS ise rastgele yürüyüş (random walk) stratejisiyle keşfedilmemiş bölgelerin bulunmasında faydalı olabilir. Bunlar geleceğe dönük öngörülerdir, ancak mevcut çalışmamız bu tür eklentiler için uygun bir platform sağlamaktadır.

**3.2.2 Gerçek Dünya Koşulları ve Sınırlılıklar**

Projemiz tamamen simülasyon ortamında gerçekleştirildiğinden, gerçek dünya uygulamasına geçerken bazı farklar olacaktır. Öncelikle, simülasyonda rüzgar, engeller, arazi yükselti farkı, GPS hatası gibi unsurlar basitleştirilmiş veya hiç modellenmemiştir. Oysa gerçekte İHA’ların uçuş stabilitesi, konum doğruluğu gibi konular bu faktörlerden etkilenecektir. Bu bakımdan, sonuçlarımızın idealleştirilmiş koşullar altında geçerli olduğu not edilmelidir. Bu bir sınırlılık olmakla beraber, ROS tabanlı yapımız sayesinde bu gerçek dünya etkilerinin eklenmesi zor değildir. Örneğin Gazebo gibi bir simülasyon motoru entegre edilerek rüzgâr kuvvetleri veya engel çarpma testleri yapılabilir. Bizim yazdığımız uav\_node.py düğümü, halihazırda sadece basit bir kinematik hareket simülasyonu yapmaktadır; eğer altına gerçekçi bir uçuş denetleyicisi modeli (PID kontrolör gibi) koymak istersek ROS buna da imkân tanıyacaktır. Dolayısıyla, bu sınırlılıklar gelecekte aşılabilir.

Bir diğer önemli nokta, gerçek drone’ların haberleşmesi için kullanılan protokollerin (Wi-Fi, LoRa, 5G vb.) performansıdır. Biz mesafeyle keskin olarak azalan bir model aldık (30 m kesip attık). Gerçekte sinyal gücü mesafe ile kademeli düşer ve ara değerler olabilir. Ayrıca parazit, çoklu yol etkisi vs. de performansı etkiler. Bu yönüyle, projemizdeki iletişim modelinin basit ama kaba olduğunu söyleyebiliriz. Buna rağmen, uç nokta senaryosu (50 m eşik) testimizde yaşadığımız kararsızlık gibi, belli gerçekçi etkileri yakaladık. Yine de, iletişim alanında uzmanlaşmış protokolleri entegre etmek, örneğin RL tabanlı anti-jamming yaklaşımı gibi, sistemimizi daha da sağlam kılacaktır. Bu da gelecekteki çalışmalar için bir öneri olarak ele alınabilir.

**3.2.3 Özgün Değerin Değerlendirilmesi**

Projenin özgün değerlerinden biri, İHA’ların arama-kurtarma operasyonlarında yalnızca veri toplamakla kalmayıp, görev koordinasyonunu otonom şekilde üstlenebilmesidir. Elde edilen simülasyon sonuçları, İHA sürülerinin koordineli hareket ederek birbirini tamamlayan görevler gerçekleştirebildiğini göstermiştir. Örneğin, PSO algoritması ile yürütülen bir senaryoda beş İHA’dan biri bölgenin kuzeyine, diğeri güneye yönelmiş; kalanlar ise merkezi bölgeleri taramıştır. Bu durum fiilen bir görev paylaşımına işaret etmektedir: Her bir İHA farklı bir alt görevi bağımsız olarak üstlenmiştir. Bu paylaşım tamamen algoritmanın optimum kapsama arayışı ile oluşmuştur; yani müdahale olmaksızın, otonom bir şekilde gerçekleşmiştir. Bu yönüyle, çalışmamız otonom görev dağılımı ve dağıtık kontrol konularında kavramsal bir katkı sunmaktadır. Literatürde benzer bir yaklaşım olarak Toorchi ve arkadaşları dinamik UAV ağları için akıllı yönlendirme algoritması olan Skeleton-Based Swarm Routing (SSR)’yi önermektedir [7]. Bizim çalışmamız ise benzer bir yaklaşımı arama-kurtarma bağlamına uygulayarak görev dağılımına dayalı bir yönlendirme stratejisi geliştirmiştir.

**3.3 Kullanım Senaryoları**

Elde ettiğimiz sonuçlar, bazı pratik senaryolarda doğrudan fayda sağlayabilecek niteliktedir. Özellikle deprem sonrası arama-kurtarma uygulamaları en belirgin kullanım alanıdır. Enkaz alanlarının havadan taranması ve yaşam belirtisi olabilecek ipuçlarının tespiti gibi görevlerde, onlarca İHA’nın koordineli çalışması hayat kurtarıcı olabilir. Bu çalışmada geliştirilen simülasyon altyapısı, bu vizyonun bir parçası olarak koordinasyon ve optimizasyon konularında başarılı bir adım sunmuştur. Diğer potansiyel senaryo ise orman yangınlarıdır. İHA sürüleri yangının yayılma yönünü ve hızını tespit edebilir; gerektiğinde yangın söndürme ekiplerine anlık bilgi aktarabilir. Bu durumda, İHA’ların ad-hoc bir iletişim ağı kurma yetenekleri öne çıkar. Zira dağlık arazide baz istasyonları zarar görebilir ve sabit altyapı çökerse, havadaki İHA'lar geçici bir iletişim altyapısı sağlayabilir. Literatürde Longyu ve arkadaşlarının çalışması [5], UAV sürülerinin bu tür iletişim görevlerinde kullanılabilirliğine odaklanmaktadır. Bizim çalışmamız da benzer şekilde, sürüyü yalnızca keşif aracı olarak değil, aynı zamanda dağıtık bir iletişim altyapısı olarak kurgulayabilmiştir. Böylece, proje yalnızca arama-kurtarma değil, aynı zamanda afet sonrası acil haberleşme konularında da katkı sunma potansiyeline sahiptir.

**3.4 Sınırlılıklar**

Projenin temel sınırlamalarından biri, simülasyon ölçeğimizin nispeten küçük ve az İHA’lı senaryolarla sınırlı kalmasıdır. Gerçekte bir kasaba ölçeğinde tarama yapmak için belki 50-100 İHA gerekebilir; bu proje kapsamında 3-7 aralığında test edildi. Bu bir hesaplama kısıtıydı ve ROS ile çok sayıda sanal İHA’yı gerçek zamanlı simüle etmek zor olabilir. Ancak algoritmalarımız kavramsal olarak ölçeklenebilirdir. PSO’nun parçacık boyutu İHA sayısıyla orantılı artacak, belki popülasyon ve iterasyon sayısı da artırılacak. Bunlar halledilebilir, ama pratik implementasyonda daha güçlü bilgisayarlar gerekecek. Dolayısıyla, bu kısıt çalışmanın genellenebilirliğini sınırlasa da, ele alınabilir.

Bir diğer kısıt, algoritmaların parametre hassasiyeti ile ilgilidir. PSO için atalet ve katsayılar, boids için komşu yarıçapı, ACO için feromon parametreleri vs., elle en iyi sonuç verecek şekilde ayarlandı. Farklı bir senaryoda bu parametrelerin yeniden ayarı gerekebilir; bu da her zaman kolay olmayabilir. Yani bizim çözümümüz bir “plug-and-play” değil de biraz ince ayar gerektiren bir yapı sunuyor. Bu, meta-sezgisel algoritmaların genel bir sorunudur aslında. Gelecekte adaptif parametre ayarlama veya öğrenen algoritma seçimi (meta-learning) ile bu kısıt aşılabilir.

**3.5 Genel Değerlendirme**

Tüm sonuçlar değerlendirildiğinde, projenin hedeflerine büyük oranda ulaştığı, ancak gerçek uygulama yolunda hala adımlar atılması gerektiği anlaşılmaktadır. Özellikle proje çıktılarının, literatürde benzer amaçlı çalışmalarla uyumlu olduğu ve bazı yenilikçi yönler getirdiği görülmüştür. İHA sürülerinin afet yönetiminde kullanımı konusunda elde ettiğimiz bulgular, hem akademik anlamda yeni sorular ortaya çıkarmakta hem de pratik uygulamalara zemin hazırlamaktadır. Sonuç bölümünde bu yönler özetlenerek, ileriye dönük yapılabilecek çalışmalar önerilecektir.

1. **SONUÇ**

Bu bitirme projesinde, afet bölgelerinde çoklu insansız hava araçlarının (İHA) etkin kullanımına yönelik algoritmalar geliştirerek simülasyon ortamında analiz ettik. Başlangıçta ortaya konan problem, mevcut İHA’ların afet durumlarında sınırlı adaptasyon gösterebilmesiydi. Bu kapsamda literatürdeki eksikleri gideren özgün bir çözüm mimarisi tasarlanmış ve uygulanmıştır.

Proje süresince, ROS tabanlı bir simülasyon sistemi geliştirilerek İHA sürülerinin koordineli çalışması sağlanmıştır. Gerçekleştirilen kapsamlı testler göstermiştir ki, uygun formasyon, iletişim menzili ve otonom görev paylaşımı algoritmaları seçildiğinde, İHA sürüleri afet yönetiminde hız ve etkinlik kazandırabilmektedir. Özellikle, önerdiğimiz PSO tabanlı dağılım algoritması ve türevleri, afet alanının hızlı taranmasında ve kritik bilgilerin toplanmasında önemli başarı elde etmiştir. Görev paylaşımı ve çarpışma önleyici sistemlerin başarısı, gerçek zamanlı karar verme yeteneklerine de bağlıdır. Zhau ve arkadaşları [13], bilişsel oyun teorisine dayalı yaklaşımlarla sürü içi çarpışma öncesi karar modellerinin daha güvenli uçuş stratejileri geliştirdiğini göstermiştir. Algoritmaların afet senaryolarına uygulanabilirliği, pratikte test edildiklerinde belirgin hale gelir. Özellikle yangın gibi hızla yayılan durumlarda sürü zekası temelli koordinasyonun etkili olduğu çalışmalarda gösterilmiştir [14]. Kod altyapımız, farklı sayılarda ve formasyon tiplerinde İHA kullanımını kolayca modelleyebilmiş ve ROS ortamında çok sayıda senaryonun hızlıca denenmesine olanak tanımıştır. Bu esnek altyapı, gerçek dünyadaki farklı afet senaryolarının çeşitliliğini karşılayabilecek bir esneklik sunmuştur.

**4.1- Dinamik Sürü Algoritması**

Geliştirilen algoritma seti, İHA’ların sürü halinde hareket ederken anlık kararlar alabilmesini, görev paylaşımını otonom yapabilmesini ve iletişim ağını korumasını sağlamıştır. Böylece özgün ve etkin bir algoritma mimarisi ortaya konmuştur.

**4.2- Simülasyon Bulguları**

Yapılan simülasyon testlerinde, önerilen yaklaşımla arama-kurtarma operasyonlarının hızlandığı ve daha verimli hale geldiği görülmüştür. İHA sürüsü kullanımı, tekil İHA kullanımına kıyasla hem kapsanan alanı artırmış hem de bilgi akışını hızlandırmıştır

**4.3- Kullanılan Teknoloji**

Sürünün ağ yapısı sayesinde, afet bölgesinde iletişim altyapısı zarar görmüş olsa bile İHA’lar kendi aralarında bir ağ kurabilmiş ve verileri başarıyla paylaşmıştır. Ayrıca optimize edilmiş rota ve dağılım sayesinde gereksiz uçuşlar minimize edilerek enerji tasarrufu sağlanmıştır.

Elbette, çalışma bazı sınırlılıklara sahiptir ve bu da gelecekte yapılabilecek çalışmalar için fırsatlar sunmaktadır. İleriye dönük öneriler ve yapılabilecek işler şu şekilde sıralanabilir:

**4.4- Gerçek Dünya Testleri**

Simülasyon ortamında doğrulanan algoritmaların, gerçek İHA platformlarında da test edilmesi kritik bir adımdır. Bu amaçla küçük ölçekli drone filoları kullanılarak kontrollü alanlarda (örn. açık bir arazide) deneyler yapılabilir. Bu sayede, gerçek sensör gürültüsü, rüzgar, iletişim gecikmeleri gibi etkenlerin algoritma performansını nasıl etkilediği gözlemlenecektir. Gerekirse algoritmalarda bu etkenlere yönelik uyarlamalar yapılabilir.

**4.5- Gelişmiş Simülasyon Özellikleri**

Mevcut simülasyona gazebo gibi fizik motorları entegre edilerek daha gerçekçi çevre modellemesi sağlanabilir. Ayrıca yangın, duman, yağmur gibi afet senaryolarına özgü unsurlar simüle edilebilir. Bu, algoritmaların bu koşullarda da çalışıp çalışmadığını test etme imkanı verir. Örneğin rüzgar esintisi, İHA’ların konumlarını bozuyorsa, algoritmanın ne sıklıkta yeniden dağılım yaptığı veya bozulmaya dayanıklı olup olmadığı incelenebilir.

**4.6- Yapay Zeka Entegrasyonu**

Karar verme süreçlerine makine öğrenmesi teknikleri entegre edilebilir. Örneğin, sürü içinden bir “lider” İHA seçilip, derin öğrenme ile enkaz tanıma modeli çalıştırarak önemli bölgeleri önceliklendirmesi sağlanabilir. Veya pekiştirmeli öğrenme ile, dağılım algoritmalarının parametreleri (PSO, ACO parametreleri gibi) dinamik olarak ayarlanabilir. Benzer şekilde, Puente-Castro ve arkadaşları, arazi taraması gibi görevlerde İHA sürülerinin yol planlamasını güçlendirmek için takviyeli öğrenme (Reinforcement Learning) yöntemlerini başarılı biçimde uygulamıştır [2].Bu sayede her yeni senaryoya adapte olan “öğrenebilen bir sürü” tasarımı ortaya çıkabilir.

**4.7- Algoritma İyileştirmeleri**

Bölüm 7’de tartışıldığı üzere, farklı optimizasyon algoritmalarının hibrit kullanımı değerlendirilebilir. Özellikle Grey Wolf veya Cuckoo Search gibi yeni popüler yöntemler, literatürde başarılı sonuçlar almıştır; bunların sürü dağılım problemimize uygulanması ilginç sonuçlar verebilir. Ayrıca halihazırda geliştirdiğimiz PSO ve ACO yaklaşımlarının parametre duyarlılıkları incelenip, adaptif parametre güncelleme mekanizmaları (ör. PSO’da atalet ağırlığının iterasyona göre azalması) eklenebilir.

**4.8- Sensör ve Veri Füzyonu**

İHA’ların üzerindeki sensörlerden (kamera, termal sensör vb.) gelen verilerin gerçek zamanlı olarak işlenip sürü kararlarına etki etmesi sağlanabilir. Örneğin bir İHA, termal kamerayla yüksek sıcaklık (yangın) algılarsa bunu diğerlerine iletip onların rotalarını buna göre yeniden planlatabiliriz. Bu tarz bir sensör veri füzyonu, sürünün çevreyi algılayıp tepki vermesini, yani durumsal farkındalığını artıracaktır.

**4.9- Operasyonel Yazılım ve Arayüz**

Geliştirilen sistemin bir üst katmanda kullanıcı arayüzü ile desteklenmesi, pratik kullanımı kolaylaştıracaktır. Örneğin bir afet koordinasyon merkezinde operatörler için, harita üzerinde İHA’ların anlık konumlarını, kapsama alanlarını gösteren ve belki manuel müdahale imkanı veren bir arayüz geliştirilebilir. Biz temel olarak arka plandaki algoritmaları kurduk; bunların insan kontrolünde izlenmesi ve gerektiğinde yönlendirilmesi için bir GUI (Graphical User Interface) önemlidir. Nitekim proje kapsamında bir prototip Qt tabanlı arayüz (sim\_gui\_final.py) hazırlanmıştır, bu ilerletilebilir.

Sonuç olarak, bu proje kapsamında afet yönetiminde kullanılacak sürü İHA’ların operasyonel etkinliğini artırmak üzere başarılı bir adım atılmıştır. Simülasyon sonuçlarımız, uygun şekilde tasarlanmış sürü algoritmalarının arama-kurtarma ve lojistik destek operasyonlarında zaman ve iş gücü tasarrufu sağlayabileceğini, dolayısıyla daha fazla hayat kurtarmaya yardımcı olabileceğini göstermektedir. İleride gerçek çevre koşullarını ve daha büyük ölçekleri de içeren çalışmalarla bu konseptin olgunlaştırılması gerektiği açıktır. Böylelikle, afet anlarında kritik rol oynayabilecek daha dayanıklı ve akıllı İHA sürüleri tasarlamak mümkün olacaktır. Bu projenin çıktıları, bu yöndeki çalışmalara sağlam bir altyapı ve deneyim kazandırmıştır.

KAYNAKLAR

[1] B. -W. Chen and S. Rho, "Autonomous Tactical Deployment of the UAV Array Using Self-Organizing Swarm Intelligence," in IEEE Consumer Electronics Magazine, vol. 9, no. 2, pp. 52-56, 1 March 2020, doi: 10.1109/MCE.2019.2954051.

[2] Alejandro Puente-Castro, Daniel Rivero, Alejandro Pazos, and Enrique Fernandez-Blanco. 2022. UAV swarm path planning with reinforcement learning for field prospecting. Applied Intelligence 52, 12 (Sep 2022), 14101–14118. https://doi.org/10.1007/s10489-022-03254-4.

[3] Degirmen, Sema & Cavdur, Fatih & Sebatli-Saglam, Asli. (2018). AFET OPERASYONLARI YÖNETİMİNDE İNSANSIZ HAVA ARAÇLARININ KULLANIMI: GÖZETLEME OPERASYONLARI İÇİN ROTA PLANLAMA. Uludağ University Journal of The Faculty of Engineering. 23. 11-26. 10.17482/uumfd.455146. [4] G. Sluder and D. E. Wolf, Digital microscopy. Academic Press, 2013.

[4] Kumar, Deepak & Sonia, Dr. (2023). Highly Optimized Energy Saving Protocol for Flying ad-hoc Network. International Journal for Research in Applied Science and Engineering Technology. 11. 2177-2189. 10.22214/ijraset.2023.49910.

[5] Peng, Jinlin & Zhang, Zixuan & Wu, Qinhao & Zhang, Bo. (2019). Anti-Jamming Communications in UAV Swarms: A Reinforcement Learning Approach. IEEE Access. 7. 180532-180543. 10.1109/ACCESS.2019.2958328.

[6] Y. Zhou, B. Rao and W. Wang, "UAV Swarm Intelligence: Recent Advances and Future Trends," in IEEE Access, vol. 8, pp. 183856-183878, 2020, doi: 10.1109/ACCESS.2020.3028865.

[7] N. Toorchi, F. Hu, E. S. Bentley and S. Kumar, "Skeleton-Based Swarm Routing (SSR): Intelligent Smooth Routing for Dynamic UAV Networks," in IEEE Access, vol. 9, pp. 1286-1303, 2021, doi: 10.1109/ACCESS.2020.3043672.

[8] G. Sun et al., "Collaborative Beamforming for UAV Networks Exploiting Swarm Intelligence," in IEEE Wireless Communications, vol. 29, no. 4, pp. 10-17, August 2022, doi: 10.1109/MWC.001.2100677.

[9] R. Arnold, E. Mezzacappa, J. Jablonski and B. Abruzzo, "Multi-Role UAV Swarm Behaviors for Wide Area Search using Emergent Intelligence," 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), London, UK, 2020, pp. 255-261, doi: 10.1109/WorldS450073.2020.9210392.

[10] Zedadra, Ouarda & Zedadra, Amina & Guerrieri, Antonio & Seridi, Hamid & Ghelis, Douaa. (2024). A New Multi-Robots Search and Rescue Strategy based on Penguin Optimization Algorithm. SCPE, 25(5). https://doi.org/10.12694/scpe.v25i5.3541.

[11] Li, Z., Bao, G., Mao, H., Jiang, M., Shi, Y., Peng, F., Cai, H. (2024). Localizing particulate matter sources in indoor environments with weak airflow: An experimental study using swarm intelligence methods. Sustainable Cities and Society, 115, 105820. 10.1016/j.scs.2024.105820

[12] Pyke, Lewis & Stark, C.. (2021). Dynamic Pathfinding for a Swarm Intelligence Based UAV Control Model Using Particle Swarm Optimisation. Frontiers in Applied Mathematics and Statistics. 7. 10.3389/fams.2021.744955.

[13] Huan Zhau, Liang Gao, and Jiyin Zhao. 2022. Anticollision Decision and Control of UAV Swarm Based on Intelligent Cognitive Game. Journal of Advanced Transportation, 2022, Article ID 6398039. <https://doi.org/10.1155/2022/6398039>.

[14] Daniel Norris. 2010. We Didn't Start the Fire! Swarm Intelligence for Forest Fire Suppression. Proceedings of the IEEE Swarm Intelligence Symposium (SIS 2010), 1–6.